3D Edge Sketch from Multiview Images

Yilin Zheng[1†] Chiang-Heng Chien[1†] Ricardo Fabbri[2] Benjamin Kimia[1]

[1]Brown University [2]Rio de Janeiro State University

Abstract

The semantic reconstruction of a scene relies in part on the curvilinear structure inherent in images. The recovery of curvilinear structure is not only key to the representation of objects via ridges and other object curves but is also critical to the reconstruction from texture-poor images which lack a sufficient number of features. Prior methods advocate for the recovery of curve segments from images and reconstructing these into an organized collection of 3D curve segments often referred to as the 3D curve sketch*, which serves as the basis for further reconstruction of curves and surfaces. Observing that the process of edge grouping can lead to fictitious curves or missing veridical groupings, this paper advocates for a reconstruction of curvilinear structure directly from image edges in the form of a* 3D edge sketch*. The multiview reconstruction of edges faces significant combinatorial challenges which are effectively addressed in this paper. We demonstrate through experiments that the 3D edge sketch recovers a vast majority of the curvilinear structure and is a reliable substrate from which 3D curves can be constructed.*

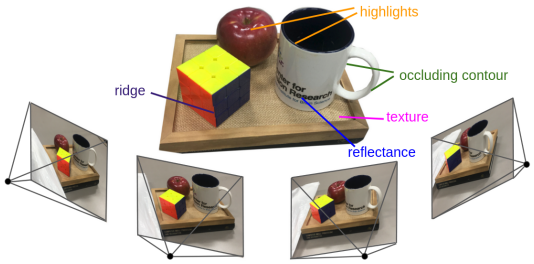

Figure 1. Images taken from multiple views are used to reconstruct view-stationary contours such as ridges, texture and reflectance edges.

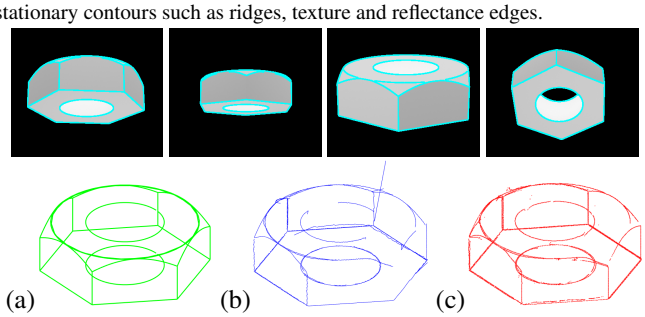

Figure 2. (**Top Row**) A few views of a 3D object with 2D edges superimposed in cyan. (**Bottom Row**) The ground-truth (GT) (a), serves as a reference to compare the 3D Curve Sketch [6] (b) to the 3D Edge Sketch (c) proposed here. The 3D Edge Sketch is richer in edge content, not suffering from gaps in the 3D Curve Sketch, and does not introduce a significant set of false positives.

1. Introduction

The 3D reconstruction of the world from a series of images has generally relied on finding correspondences among point features in multiple views and triangulation of these correspondences into an unorganized cloud of 3D points [22]. A significant drawback of this approach is that some scenes have featureless surfaces or surfaces lacking a dense collection of point features. At other times, there is a dense collection of 3D points, but there is repeated textures or a large proportion of outliers, say due to a wide baseline, such that a 3D reconstruction via feature points is not viable. Furthermore, even if a 3D set of points can be successfully reconstructed, it may not be sufficiently dense to be organized as a mesh. Finally, even if a mesh can be constructed, such a representation does not lead itself to a semantic interpretation. For example, the mesh representation of a chair does not organize points into surface patches corresponding to the seat, the back, *etc*. As a more concrete example, consider viewing a cube from multiple directions. Even if the cube is densely textured, at best it is reconstructed as a dense collection of 3D points that are meshed as a single surface, often smoothed around ridges: there is no notion of 6 faces meeting at 12 ridges and 8 vertices.

A curve-based reconstruction can often address these shortcomings and complement the point-based reconstruction: *(i)* when point features are lacking, there are often ample curve segments, *(ii)* curves are more stable to illumination and view changes, *(iii)* a curve-based reconstruction represents surface ridges so that it provides a scaffold on which surface reconstruction can be organized into semantically meaningful surface patches, *e.g.*, for the chair or

† Equal contribution.

cube examples considered earlier.

Fabbri *et al.* [6] pioneered multiview curve-based reconstruction in the form of a *curve sketch* which is an unorganized collection of 3D curves, and later as a curve *drawing* [30] which is a graph of 3D curves. Surfaces are then reconstructed by connecting curves using multiview lofting and enforcing multiview consistency [31]. The approach begins with extracting 2D curve fragments from each image by edge detection and edge grouping. This process critically relies on an orientation-corrected third-order edge detection [16] which gives reliably localized subpixel edge position and edge orientations, followed by a reliable grouping first as *curvels* and then as *curve fragments* [9]. Each curve fragment in each view then seeks a pairing with other curve fragments falling in epipolar bands from other views, and each pairing constitutes a hypothesis. The hypothesis is effectively a 3D curve fragment that can be reprojected to other views. Each 3D curve hypothesis seeks support from edge content in each projected view. The majority of the hypotheses are incorrect and do not find such support, where the veridical hypothesis gathers significant support and gets verified as a member of the 3D curve sketch.

The above process relies on the correct grouping of edges into curves from which a sketch is built. The question then arises whether it is possible to form a 3D hypotheses *without grouping*, by directly pairing 2D edge elements *across* pairs of views to form 3D edges hypotheses and verify them by projecting on other views? If this is possible, then the reconstructed 3D edge can be grouped into 3D curves in 3D space. The conjecture of this paper is that by directly working with edges, and not curves, the process does not rely on the correctness of the 2D edge grouping. The drawbacks of such an approach, however, is that *(i)* the number of edge pairs considered is significantly higher, and *(ii)* there are significantly more outliers, since the cluttered edges which are filtered out in an edge grouping process must be dealt with. This paper explores the direct reconstruction of 2D edges into an unorganized set of 3D edges, referred to as the **3D Edge Sketch**. The resulting reconstruction could then serve as a sound basis for a more globally consistent grouping of 3D edges into 3D curves. Figure 2 illustrates the contribution. For the benefit of the research community, the code is made publicly available at https://github.com/C-H-Chien/3D_Edge_Sketch.

2. Related Works

Beyond 3D curve sketch and drawing [6, 30, 31], reconstructing edges, lines, or curves have been motivated early-on by the need to reconstruct thin structures, *e.g.*, wire frames, art, and hairs [4, 13, 14, 18, 20, 24, 27, 32]. They typically first construct a cloud of 3D points, followed by extracting curves from a curve bundle [23] or clustered line segments [27]. Also, 3D edge reconstruction of an object has also become an important problem as 3D sharp, geometric edges can be used as an abstraction of a complex object shape, facilitating tasks such as surface reconstruction [7, 31] and shape reconstruction [25]. Given a 3D point cloud, many methods extract edges using geometric features, *e.g.*, normals and curvatures [10, 15, 21, 35, 38], others formulate 3D edge detection as a classification problem based on the edge neighborhood [2, 12, 37], or learn parametric curves from 3D edges given point clouds [5, 24, 34]. NerVE [39] learns 3D edges from a point cloud by representing edges as grids of volumetric cubes converted to piece-wise linear curves. These methods give promising results, yet are very sensitive to noise and non-uniform density of a point cloud arising from noisy observations such as real images. This motivates another stream of 3D edge reconstruction: rather than acting in a *representation space* of 3D points, reconstructing 3D edges from *observation space* of images avoids the reliance on a perfect point cloud.

Semantic segmentation has been used on images from vehicles to estimate ground surface, constructing 3D edge road maps arising from 2D semantic contours [11]. Similarly, others construct indoor topological contour maps [33]. These methods are incremental and require an ordered sequence of images. Multivew stereo 3D edge reconstruction [3] triangulates 2D edge correspondences to 3D edges from a set of unordered images. The 2D edge correspondences are hypothesized and verified through spatial consistency with 3D point cloud and epipolar-geometric supports. It has shown promising qualitative results against OpenMVG [26] when used for constructing meshes, but an initial point cloud is necessary. Also, edge orientation which is a strong cue, is ignored, leading to false correspondences. Deep learning methods such as Neural Edge Field (NEF) [36] create 3D parametric edges from multiple 2D edges, but are limited to simple objects, *e.g.*, cubes or chunks [17], often require scene specific training and careful preparation of the dataset is necessary for training, the network. Another recent learning-based 3D edge reconstruction, namely, EMAP [19], showed significant improvement over NEF for both synthetic, simple objects and real, large complicated scenes. Nevertheless, it also requires scene specific training, *i.e.*, its model trained on one scene can not be generalized to another scene.

e	Edge, $e = (\gamma, \theta)$
e_i^n	Edge i in view n
γ	An image location $\gamma = (\xi, \eta, 1)^T$
θ	orientation $0 \leqslant \theta < \pi$
Γ	A 3D point, $\Gamma = (X, Y, Z)$ in world coordinate
Γ^n	A 3D point in the n^{th} camera coordinate system $\Gamma^n = \mathcal{R}_n\Gamma + \mathcal{T}_n$
T_n	A 3D unit orientation vector in the n^{th} camera coordinate system
$(\mathcal{R}_n, \mathcal{T}_n)$	Rotation matrix and translation vector of the n^{th} camera
$(\mathcal{R}_{ij}, \mathcal{T}_{ij})$	$\mathcal{R}$ and $\mathcal{T}$ relating i^{th} camera to j^{th} camera, $\Gamma^i = \mathcal{R}_{ij}\Gamma^j + \mathcal{T}_{ij}$
ρ	Depth of a point $\Gamma = \rho\gamma$
$\mathcal{C}_n^i$	The i^{th} curve segment in the n^{th} image

Table 1. Notations used in this paper.

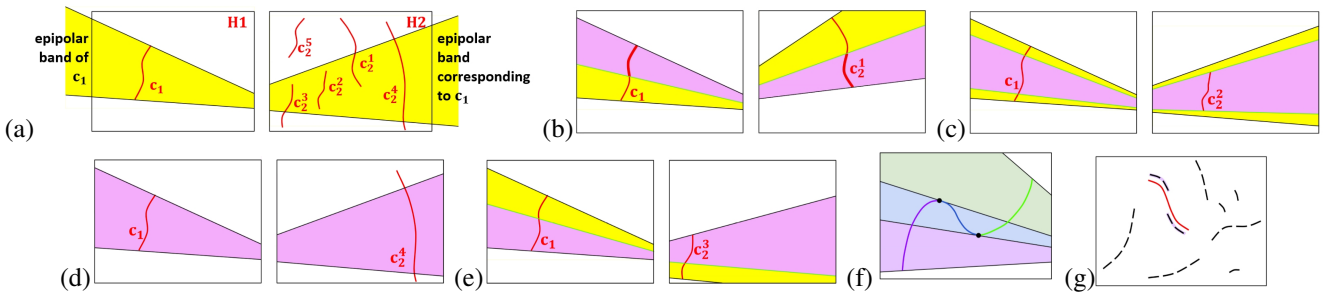

Figure 3. (a) Curve segments sharing a common epipolar band are paired as a 3D curve hypothesis in the 3D curve sketch [6], *i.e.*, $\mathcal{C}_1$ is paired with $\mathcal{C}_2^1$, $\mathcal{C}_2^2$, $\mathcal{C}_2^3$, $\mathcal{C}_2^4$, but not $\mathcal{C}_2^5$. (b,c,d,e) Only curve subsegments in common epipolar bands (pink) are paired. (f) Curve must be broken into segments at epipolar tangencies. (g) A projected curve (red) seeks support from edges.

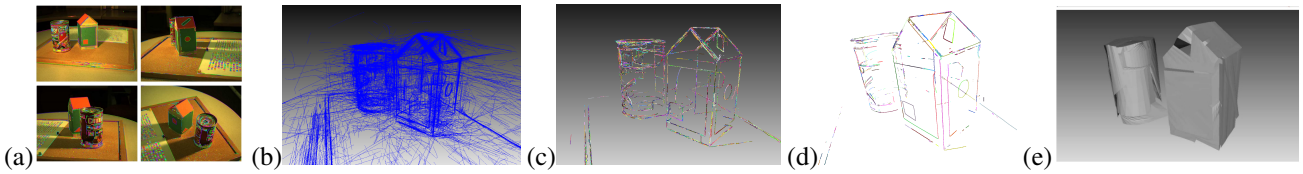

Figure 4. From [30, 31]: (a) four selected views of a scene, (b) 3D curve sketch, (c) enhanced curve sketch, (d) 3D drawing, and (e) surface reconstruction.

3. 3D Curve Sketch vs 3D Edge Sketch

Fabbri and Kimia [6] presented a general approach for finding corresponding curve segments and reconstructing a cloud of 3D curve segments, *the 3D curve sketch*. Their approach uses the epipolar constraint to find potential matches for a curve segment in one view with other curve segments in other views. Figure 3 illustrates this approach with an example where a curve $\mathcal{C}_1$ in hypothesis view H_1 explores pairing with curves in the corresponding cpipolar band in the second hypothesis view H_2, *i.e.*, $\mathcal{C}_2^1$, $\mathcal{C}_2^2$, $\mathcal{C}_2^3$, and $\mathcal{C}_2^4$. Consider a potential pairing of $\mathcal{C}_1$ and $\mathcal{C}_2^1$ where the common epipolar band delineates a segment of $\mathcal{C}_1$ and $\mathcal{C}_2^1$ that have a set of corresponding points. These two sub-segments of $\mathcal{C}_1$ and $\mathcal{C}_2^1$ are then triangulated to form a 3D curve segment hypothesis, which may or may not be correct, pending support from other views. When curve segments become tangential to the epipolar line, they must be divided into segments and treated distinctly, Figure 3(f).

The vast majority of 3D curve segment hypotheses are not veridical. The veridical pairings can be differentiated from non-veridical pairings by probing additional views. Each 3D curve hypothesis is projected onto other views (validation views) where evidence for it is sought in the form of consistency with the curve structure, Figure 3 (g). Since curve grouping can often group unrelated edges, or not group edges that do lie on a curve, validation evidence is accumulated directly from the edge map by counting the number of edges supporting a projected hypothesis curve, Figure 3(g). Generally, four validation views with sufficient edge evidence suffice to validate a 3D curve segment.

Subsequent work [30] addresses several shortcomings: gaps resulting from epipolar breakup of curves, redundancy arising from multiple hypothesis representing the same 3D curve, *etc*. The resulting enhanced curve sketch is referred to as a *3D curve drawing*, Figure 4, which is the basis of generating surfaces, Figure 4(e).

The current paper similarly aims at reconstructing 3D curve structure, but it takes a drastically different approach: The approach in [6] critically relies on the grouping of edges into curve segments. It is well-known, however, that each grouping can join unrelated edges at times, or leave behind gaps of ungrouped edges which are part of the same curve. In other words, while [6] produces meaningful 3D curve structure, it also inherits the drawbacks of the 2D edge grouping process without recourse to correcting these. It thus leaves behind both gaps and 3D curves which could have been reconstructed, but are not due to inappropriately grouped edges or ungrouped edges.

This paper proposes to switch the order of grouping and reconstruction by first reconstructing 3D edges and then grouping these to produce 3D curve segments. The process of reconstructing 3D edges to form a *3D Edge Sketch*, however, is quite challenging: an advantage of grouping before reconstruction is that the thousands of edges are reduced to dozens of curve segments. Each curve segment may intersect with a dozen curve segments in its epipolar band and then face the counting of edges in the vicinity of projected hypotheses. In contrast, each edge will be paired with significantly more edges, both those that are veridical and those which are not. An edge will also pair with multiple close-by edges representing the same curve, leading to highly redundant hypotheses. Furthermore, in the validation process, each 3D edge hypothesis will have to consider many edges in a neighborhood of the projected 3D edges due to errors in edge localization and poses. As a result, or-

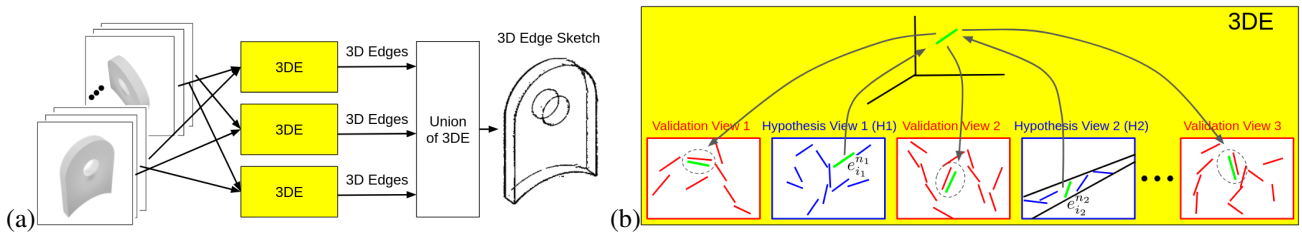

Figure 5. The pipeline of 3D Edge Sketch: (a) Edges from pairs of views are reconstructed and their union generates the 3D Edge Sketch. (b) The reconstruction of 3D edges from two hypothesis views H_1 and H_2 iterates over all edges of H_1 pairing with all edges of H_2 in the corresponding epipolar wedge to form 3D edge hypothesis. These hypotheses are verified only when a sufficient number of validation views show supporting edges consistent with the projection of the 3D edges on that view.

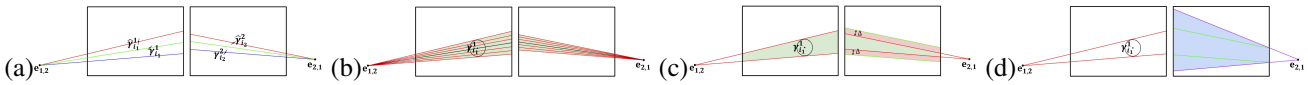

Figure 6. (a) A pair of corresponding true edges $(\hat{\gamma}^1_{i_1}, \hat{\gamma}^2_{i_2})$ are perturbed to observed locations $(\gamma^1_{i_1}, \gamma^2_{i_2})$, which may not necessarily be on corresponding epipolar lines. (b) $\hat{\gamma}^1_{i_1}$ lies within a circle of radius Δ at $\gamma^1_{i_1}$ giving rise to a pencil of epipolar line bounded by tangents to the circle. This wedge of epipolar lines corresponds to a wedge of epipolar lines in image 2, where $\hat{\gamma}^2_{i_2}$ must lie. (c) The corresponding wedge must to dilated by Δ to account for perturbation of $\gamma^2_{i_2}$. (d) An approximation of this region by an epipolar wedge. Region sizes are magnified for illustration purposes.

ders of magnitude more hypotheses are generated, Figure 5.

There is a silver lining to such a combinatorial nightmare, however: *(i)* all processes are independent and can be done in parallel, leaving the possibility of parallel computation and the use of GPUs, which is leveraged in this paper, *(ii)* the 3D edge hypotheses are consistent across views while 3D curve hypotheses are not: When using curve segments, different hypothesis pairs of views generate different groupings, so that 3D curve hypotheses are not consistent across different selections of hypotheses views. When using edges, however, an edge that validates a 3D edge hypothesis will no longer need to be considered in forming an alternate hypothesis as they are already tagged as reconstructed. As the number of tagged edges increases, the number of free edges available for forming a 3D edge hypothesis rapidly decreases, thus taming the combinatorics.

The advantage of forming a 3D edge sketch directly from a 2D edge is clear: *(i)* the process is immune to 2D edge grouping errors, *(ii)* the subsequent 3D grouping of 3D edges into 3D curves (not considered in this paper) can rely on multiple view grouping cues. This type of multiview edge grouping into curves has never been tried before; the 3D edge sketch provides a solid basis for that. The comparison of ungrouped 3D edges to the 3D curve sketch in Figure 2 clearly shows that *(i)* gaps evident in the 3D curve sketch are not present in the 3D edge sketch, *(ii)* the false positive edges in the 3D edge sketch are comparable to the 3D curve sketch even without the benefit of grouping.

4. Formation of a 3D Edge Hypothesis

The first task is to form 3D edge hypotheses. Consider a set of views I^n, $n = 1, 2, ..., N$ and let e^n_i, $i = 1, 2, ..., M_n$ represent the M_n edges in the n^{th} view, where each edge is represented as $e = (\gamma, \theta)$, where $\gamma = (\xi, \eta, 1)^T$ is its location and $0 \leqslant \theta < \pi$ is its orientation, Table 1. Note that orientation is typically not used particualrly due to the lack of orientation estimation, *e.g.*, PiDiNet [29] and DexiNet [28]. However, edge orientation is key to multiview validation. Furthermore, edge orientation is a strong grouping cue and should not be dismissed. As such, we use a third-order edge detection scheme [16] which provides highly accurate and reliable edge orientation, and a dense collection of highly accurate edges so that there are typically several edges, typically 1-3 edges, per pixel. This process leads to a rich pool of dense, spatially accurate, 2D edges per pixel with accurate orientations.

The formation of a 3D edge hypothesis begins with selecting an edge $e^{n_1}_{i_1}$ in H_1 and then exploring possible groupings with other edges $e^{n_2}_{i_2}$ in H_2, Figure 6. A pair of corresponding edges $(e^{n_1}_{i_1}, e^{n_2}_{i_2})$ must lie on corresponding epipolar lines. However, two types of errors, namely, edge localization error and relative pose error, perturb this relationship.

First, consider the *edge localization error*, where the true edge $\hat{e} = (\hat{\gamma}, \hat{\theta})$ is observed under perturbation as $e = (\gamma, \theta)$. A simple model of perturbation limits $|\hat{\gamma} - \gamma| < \Delta$, say $\Delta = 0.3$ pixels, *i.e.*, the true location of an observed edge is within a circle of radius Δ pixels. More specifically, consider a pair of true edge correspondences $(\hat{\gamma}^1_{i_1}, \hat{\gamma}^2_{i_2})$ which are observed under perturbation as $(\gamma^1_{i_1}, \gamma^2_{i_2})$. Figure 6(a) observes that while $(\hat{\gamma}^1_{i_1}, \hat{\gamma}^2_{i_2})$ are on the corresponding epipolar lines (shown in red), $(\gamma^1_{i_1}, \gamma^2_{i_2})$ do not necessarily lie on corresponding epipolar lines, shown in green and blue, respectively. Since the location of $\hat{\gamma}^1_{i_1}$ is not known, but restricted to lie within a circle of radius Δ from $\gamma^1_{i_1}$, this generates a pencil of epipolar lines which together with their corresponding counterparts are shown in Figure 6(b). This defines an *epipolar wedge*, Figure 6(c) left, and its corresponding epipolar wedge, Figure 6(c) right, repre-

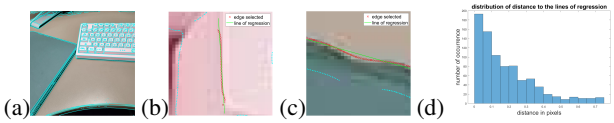

Figure 7. Estimating spatial errors Δ: Edges of an image (a) taken along linear curves shown in (b, c) are fit with a line and distribution of edge distances to this line are plotted, (d) suggesting an estimate of Δ of $\sim$0.3 pixels.

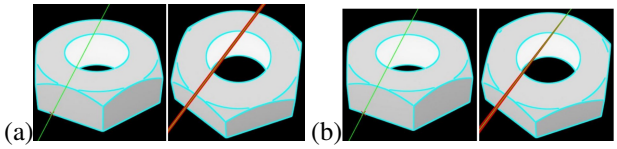

Figure 8. Perturbation of the epipolar correspondence between H_1 and H_2 is shown for (a) $\Delta\mathcal{R} = 0.25\%\angle\mathcal{R}$ and (b) $\|\Delta\mathcal{T}\| = 0.5\%\,\|\mathcal{T}\|$.

senting the region where $\hat{\gamma}^2_{i_2}$ must lie. Since $\hat{\gamma}^2_{i_2}$ is also experiencing perturbation of its own Δ, the wedge must be dilated by Δ and this region is where $\gamma^2_{i_2}$ must be sought, Figure 6(c), namely, the union of green and red regions. Since epipolar lines can easily be identified with an epipolar angle, an epipolar wedge is identified by an epipolar angle interval and similarly the corresponding epipolar wedge. Thus, it is useful to approximate this region with the smallest epipolar wedge that contains its region, Figure 6(d), shown in blue. This is typically found by the two epipolar lines going through the boundaries of the exact region. Note that these regions are in fact much smaller and have been magnified here for illustration purposes.

The approximate region helps with efficient indexing to identify edges which can potentially pair with the first edge in H_1. Once these edges are identified, however, their membership in the exact region identified in Figure 6(c) is tested. The value of Δ is experimentally determined from images with long linear edges: The detected edges are fit with a line minimizing least square errors and the distribution of the distance and orientation error with respect to this line is obtained, Figure 7.

Second, pose perturbation also leads to the perturbation of the expected location of a corresponding feature. Specifically, the perturbation of the true pose $(\hat{\mathcal{R}}, \hat{\mathcal{T}})$ is $(\mathcal{R}, \mathcal{T})$ where $\mathcal{R} = \Delta\mathcal{R}\hat{\mathcal{R}}$ and $\mathcal{T} = \hat{\mathcal{T}} + \Delta\mathcal{T}$. The perturbation of pose $(\Delta\mathcal{R}, \Delta\mathcal{T})$ has six degrees of freedom, but they are generally quantified by only two numbers, namely, $\angle\Delta\mathcal{R}$ the turn angle of $\Delta\mathcal{R}$ in an angle-axis representation, and $\|\Delta\mathcal{T}\|$, the magnitude of $\Delta\mathcal{T}$, Figure 8.

This process results in the formation of a 3D edge hypothesis formed by pairing 2D edges. Figure 9 shows an example of the number and distribution of pairings for an edge selected in H_1, with edges falling in the corresponding epipolar wedge in H_2.

5. Validating 3D Edge Hypothesis

The validation of a 3D edge hypothesis formed from a pair of edges $e^{n_1}_{i_1} = (\gamma^{n_1}_{i_1}, \theta^{n_1}_{i_1})$ and $e^{n_2}_{i_2} = (\gamma^{n_2}_{i_2}, \theta^{n_2}_{i_2})$ is

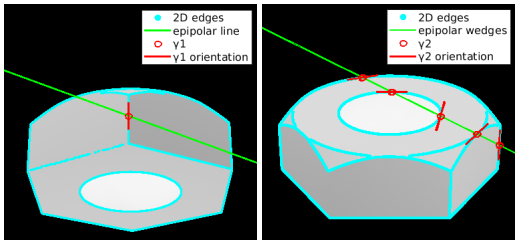

Figure 9. An edge from one hypothesis image (**Left**) pairs with edges falling within the corresponding epipolar wedge in a second hypothesis view (**Right**).

based on the consistency of its projection with the edge in the validation views. It should be noted that it is not necessary to explicitly form a 3D edge hypothesis. Rather, the edge pairs $(e^{n_1}_{i_1}, e^{n_2}_{i_2})$ are directly mapped to a validation view as $(e_3(\gamma_3, \theta_3))$, as shown below with proofs provided in the supplementary material.

Proposition 1. *Let an edge $e_1 = (\gamma_1,\ \theta_1)$ in image 1 correspond to an edge $e_2 = (\gamma_2,\ \theta_2)$ in image 2 where the relative pose of the camera is $(\mathcal{R}_{21},\ \mathcal{T}_{21})$. Then the corresponding edge in a third view $e_3 = (\gamma_3,\ \theta_3)$ with relative pose to camera 1 $(\mathcal{R}_{31},\ \mathcal{T}_{31})$ is given by*

$$\gamma_3 = \frac{\begin{array}{r}\left[\left(b_3^T T_{21}\right)\left(b_3^T \mathcal{R}_{21}^T \gamma_2\right) - \left(b_3^T \mathcal{R}_{21}^T T_{21}\right)\right] \mathcal{R}_{31}\gamma_1 \\ + \left[1 - \left(b_3^T \mathcal{R}_{21}\gamma_1\right)\left(b_3^T \mathcal{R}_{21}^T \gamma_2\right)\right] T_{31}\end{array}}{\begin{array}{r}\left[\left(b_3^T T_{21}\right)\left(b_3^T \mathcal{R}_{21}^T \gamma_2\right) - \left(b_3^T \mathcal{R}_{21}^T T_{21}\right)\right]\left(b_3^T \mathcal{R}_{31}\gamma_1\right) \\ + \left[1 - \left(b_3^T \mathcal{R}_{21}\gamma_1\right)\left(b_3^T \mathcal{R}_{21}^T \gamma_2\right)\right]\left(b_3^T T_{31}\right)\end{array}}. \tag{1}$$

$$t_3 = \frac{\begin{array}{r}\mathcal{R}_{13}\left[(\gamma_1 \times t_1) \times \mathcal{R}_{21}^T (t_2 \times \gamma_2)\right] \\ - b_3^T\left[(\gamma_1 \times t_1) \times \mathcal{R}_{21}^T (t_2 \times \gamma_2)\right]\gamma_3\end{array}}{\begin{array}{r}\left\|\mathcal{R}_{13}\left[(\gamma_1 \times t_1) \times \mathcal{R}_{21}^T (t_2 \times \gamma_2)\right]\right. \\ \left. - b_3^T\left[(\gamma_1 \times t_1) \times \mathcal{R}_{21}^T (t_2 \times \gamma_2)\right]\gamma_3\right\|\end{array}}, \tag{2}$$

where $b_3 = [0, 0, 1]^T$ and $t_i = [\cos(\theta_i),\ \sin(\theta_i),\ 0]^T$.

Note that the expected location of validation edges is the intersection of the two perturbation regions bounded by green lines in figure 6(d), one from each view, Figure 10. Any edges within his epipolar quadrilateral in green are consistent in location with the 3D edge hypothesis. Orientation of the validating edge must also be within a threshold of $\Delta\theta = 15^\circ$ of the projected 3D edge hypothesis orientation. The presence of a single validating edge is sufficient to count that view as a validating view. Each 3D edge hypothesis with a minimum of $M = 4$ view is considered as validated. This approach presents three significant challenges that must be addressed.

Combinatorial Explosion: Multi-level Thresholding: The process of forming 3D edge hypotheses generates a large number of edges, the product of the number of edges N and the number of edges within the epipolar wedges N_1, which is itself proportional to N. A key observation to reducing the N^2 nature of growth of hypotheses is that image edges arising from the same source typically, but not

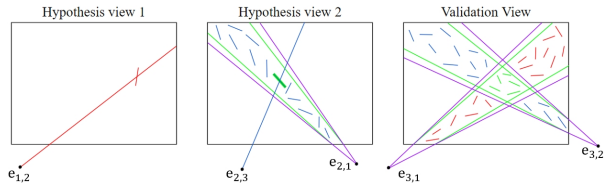

Figure 10. A selected edge in H_1 can correspond to any edge shown in blue, in the green area of H_2 and in the validation view V_1. Consider any selected edge in H_2, shown in green. The corresponding edges in V_1 to this edge are shown red edge in V_1. Thus, the intersection of blue and red edges in V_1 shown in green, correspond to the edge pair. Edges whose orientation is also consistent verifies the hypothesis.

always, have similar edge contrast. Initially only edges matching the same constrast level are considered. All edges that are validated and confirmed as arising from the same 3D edge hypothesis are removed from the pairing process. The process is then repeated with a lower but required level. The advantage, say for an eight-step contrast subdivision is that typically $N/8$ edges are paired with $N_1/8$ edges in eight steps, thus reducing the computation significantly.

Combinatorial Explosion: Parallelization: While some parts of the hypothesis construction are incremental or sequential, the consideration of each edge in H_1 in the process of forming hypotheses as well as the validation process over many views are independent and thus present valuable opportunities for parallelization. Using all the scenes of a ABC-NEF dataset [36], we have experimented both with a CPU multi-threading execution and with a GPU implementation. On average, we found that using a 32-core CPU leads to a speed up of 9.69×(∼34.3s) while using an V100 GPU led to a speed up of 617.7× (∼55.4ms) for reconstructing edges of a pair of hypothesis views and 48 confirmation views, with about 3K edges each, thus putting the process into practical range. Note that the recent deep learning approaches, namely, NEF [36] and EMAP [19], running on V100 GPU require on average 4.77 seconds and 5.69 seconds for all scenes in the ABC-NEF dataset, respectively, where each scene has to be individually trained using 3-6 hours.

Combinatorial Explosion: Epipolar Angle Indexing: One approach in addressing combinatorial explosion is to make the process more efficient. A key idea is to index edges by their epipolar angles pair in the combinatorial process so that edges within an epipolar wedge can be quickly identified. Specifically, for each pair of views, the epipolar angle of each edge in each of the two views is calculated and sorted in an ascending order. Each epipolar angle θ_1 in image 1 transforms to epipolar angle θ_2 in image 2 determined by

$$\tan(\theta_2) = -\frac{f_{11} + f_{12}\tan(\theta_1)}{f_{21} + f_{22}\tan(\theta_1)}, \tag{3}$$

where f_{ij} are the elements of the fundamental matrix F_{12}. Details are given in the supplementary material.

6. Overall Framework

The previous sections showed how 3D edge hypotheses can be formed from edges in a pair of hypothesis views and then validated based on other views. It remains unclear, however, that given $\binom{N}{2}$ pairs of N views, which pairs of views must be considered and in what order. First, when a 3D edge is observable in M views, there are $\binom{M}{2}$ distinct hypotheses, representing a high degree of redundancy. Such redundancy can be reduced by sampling views. However, certain edges might be occluded so that they only appear in select views, preventing an apriori determined order of sampling the views. Second, recall that edge pairs from a narrow baseline pair of images cannot be stably reconstructed, nor stably remapped onto other views. Thus, the approach adopted here is an iterative one: the image pair with the widest baseline is selected first and all the 3D edge hypotheses in this pair are formed and validated. Those edges in the validation views which support these 3D edge hypotheses are then tagged as belonging to this hypothesis and prevented from generating their own hypotheses when the validation view changes role to a hypothesis view in later iterations. Of course if this edge is only coincidentally supporting a hypothesis, there is a danger of missing valid 3D edge hypotheses arising from it. However, the expectation is that each 3D edge is visible in many views so that one inappropriate dismissal is not catastrophic.

Once all the 3D edge hypotheses from the initially selected pair of views are considered, a second pair of views is selected. The selection favors images that still retain a large number of untagged, available edges as well as image pairs that have a large baseline. The process is repeated for this image pair and a third image pair is next selected. The process continues until 90% of the 2D edges in each of the views are tagged as either forming a 3D edge hypotheses or validating one. The result is an unorganized cloud of 3D edges, each of which knows which two image edges generated it as a hypothesis and which edges validated it. It is noteworthy that while the position and orientation of each 3D edge is reconstructed only from the two views that formed its hypothesis, there is remarkable alignment among neighboring edges. We are currently exploring multiview triangulation [8] of edges from all the edges that formed and validated each hypothesis for an even better alignment, one which can readily support 3D grouping of these 3D edges.

7. Experiment Results

The experimental results aim to demonstrate that 3D edge sketch captures more of the 3D curvilinear structure than the 3D curve sketch, while only minimally increasing false positives. Increased recall without increased false positive rate would validate the idea of reversing the order of grouping and 3D reconstruction. Such a high-recall array

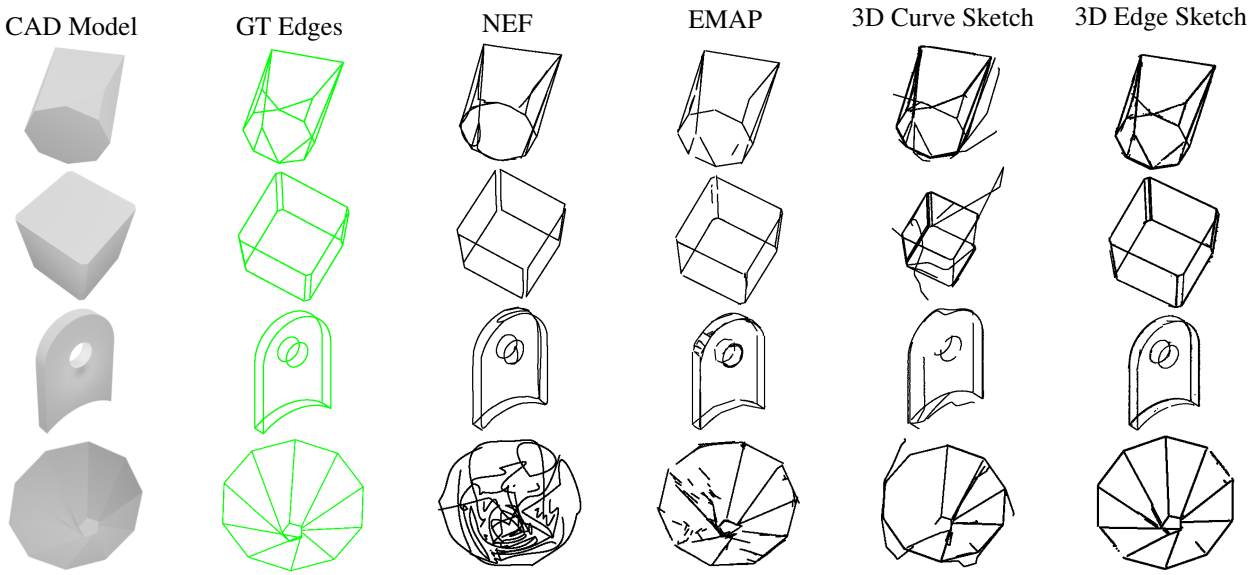

Figure 11. Qualitative comparisons between NEF [36], EMAP [19], 3D curve sketch [6], and 3D edge sketch on the ABC-NEF dataset [17]. 3D edge sketch provides a clean, complete, and natural reconstruction compared to other existing methods. More results can be found in the supplementary materials.

Figure 12. Qualitative comparisons between NEF [36], EMAP [19], 3D curve sketch [6], and 3D edge sketch on the DTU dataset [1].

of 3D edges would then serve as a substrate for 3D grouping into 3D curves.

Datasets. Both synthetic dataset ABC-NEF [17, 36] and real world dataset DTU [1] are used for evaluations. Specifically, the ABC-NEF dataset provides 115 textureless synthetic CAD models, each imaged from 50 views together with parametric ground-truth curves and edges. The DTU dataset provides scans of objects with 48-63 views. We selected six scenes containing various DTU objects for our experiments.

Baselines Approaches. Aside from the 3D curve sketch, the proposed 3D edge sketch is compared with the state-of-the-art learning-based approaches, *i.e.*, NEF [36] and EMAP [19]. NEF gives 3D parametric curves while EMAP provides 3D edges. In this comparison, the most favorable settings for these approaches are used: the third-order edge detection [16] is used for curve sketch, and PiDiNet [29] and DexiNet [28] are used for 2D edge detection for NEF and EMAP, respectively. Since the PiDiNet [29] and DexiNet [28] edge detection methods do not provide edge orientation, we opt to use the third-order edge detection. We train NEF and EMAP from scratch using their suggested parameters for all scenes. The hypothesis views and confirmation views of the curve sketch process is the same as that of the edge sketch. Note that we do not employ [3] for comparisons using ABC-NEF dataset because [3] requires an initial point cloud as its input, and ABC-NEF dataset hosts textureless objects which render no initial point cloud for evaluating [3].

Metrics. For quantitative comparisons using the ABC-NEF dataset, we follow the metrics used in [19,36], *i.e.*, Accuracy (Acc), Completeness (Comp) in millimeters, Precision (P_τ), Recall (R_τ), and F-score (F_τ) in percentage with a threshold τ for comprehensive evaluations.

Qualitative Evaluations. Figure 11 and 12 provide a qualitative comparison of four approaches on examples from the ABC-NEF and the DTU datasets, respectively. Observe that there are significant gaps in all methods except 3D Edge Sketch. In addition, in 3D curve sketch, some curves which were entirely absent are now visible. Furthermore, NEF and EMAP present noisy curves/edges while 3D edge sketch is mostly clean. Figure

The Role of Hyperparameters. An important question is

Figure 13. 3D edge sketch results without (**Left**) and with the orientation constraint (**Right**).

Methods	Representation	Acc(↓)	Comp(↓)	$P_5(\uparrow)$	$P_{10}(\uparrow)$	$P_{20}(\uparrow)$	$R_5(\uparrow)$	$R_{10}(\uparrow)$	$R_{20}(\uparrow)$	$F_5(\uparrow)$	$F_{10}(\uparrow)$	$F_{20}(\uparrow)$
NEF	Curves	14.2	15.3	7.2	52.3	87.2	16.0	66.6	94.1	9.7	57.7	89.6
EMAP	Edges	**9.4**	9.3	43.1	82.7	93.4	**47.2**	79.4	91.3	**51.7**	85.9	91.9
3D Curve Sketch	Curves	19.9	17.5	45.7	84.9	94.0	9.7	33.3	69.4	15.8	49.2	78.9
3D Edge Sketch	Edges	11.4	**4.8**	**57.6**	**90.3**	**95.8**	42.9	**82.3**	**94.4**	49.2	**88.0**	**93.6**

Table 2. Quantitative comparisons between NEF [36], EMAP [19], 3D Curve Sketch [6], and 3D Edge Sketch on the ABC-NEF dataset [17].

Figure 14. Comparison of 3D edge sketch results with different values of Δ, for (**Left**) $\Delta = 0.1$ pixels, (**Middle**) $\Delta = 0.3$ pixels, and (**Right**) $\Delta = 1.0$ pixels, while $\Delta\theta = 15^\circ$ and $N = 4$ views.

whether the method is sensitive to the choice of hyperparameters, namely, the accuracy of features Δ, the accuracy of orientation $\Delta\theta$, and minimum number of confirmation views N. Figure 13 qualitatively demonstrates the critical significance of employing orientation in the process, while Figure 14 qualitatively demonstrates the effect of varying Δ.

Figure 15 explores performance for $\Delta \in \{0.05, 0.08, 0.1, 0.3, 0.6, 1.2\}$ pixels, $\Delta\theta \in \{1^\circ, 2^\circ, 5^\circ, 10^\circ, 15^\circ, 30^\circ\}$, and $N \in \{4, 6, 8, 10, 12, 16\}$. The approach is to keep two of the three parameters fixed and use the third as the latent variable generating a precision-recall curve. Specifically, the first row selects $\Delta = 0.3$ and $N = 4$, while $\Delta\theta$ varies and this generates the green curve. Similarly, each setting of N creates a different colored curve. The second row sets $\Delta\theta = 15^\circ$ and sets Δ different values in turn while for each fixed setting, N is the latent variable. Finally, the third row sets $N = 4$ and sets Δ in its various options while $\Delta\theta$ is the latent variable. The immediate impression is that precision is high in all settings (left column), so much so that we have to zoom in to observe the variations (right column). The second impression is that recall is highly variable: the first row suggests that the highest recall is achieved with $N = 4$ but with some drop in precision, although $N = 6$ and $N = 8$ are not far behind. The second row suggests that the best recall is achieved with $\Delta = 1.2$ but $\Delta = 0.6$ and $\Delta = 0.3$ are not far behind and they have better precision. Similarly, the last row confirms a similar conclusion. Overall, Figure 14 suggests a wide basin of appropriate hyperparameters for the system. Our fixed choice of parameters for all experiments are $\Delta = 0.3$ pixels, $\Delta\theta = 15^\circ$, and $N = 4$ views.

Quantitative Evaluations. The ground-truth (GT) curves and edges are available for the ABC-NEF dataset, allowing for a quantitative comparison with competing approaches. Table 2 summarizes accuracy (Acc), completeness (Comp) in mm, and precision (P_τ), recall (R_τ), and F-score (F_τ) in percentage where τ is the threshold. This table shows that our method generally performs better although in some cases EMAP does better.

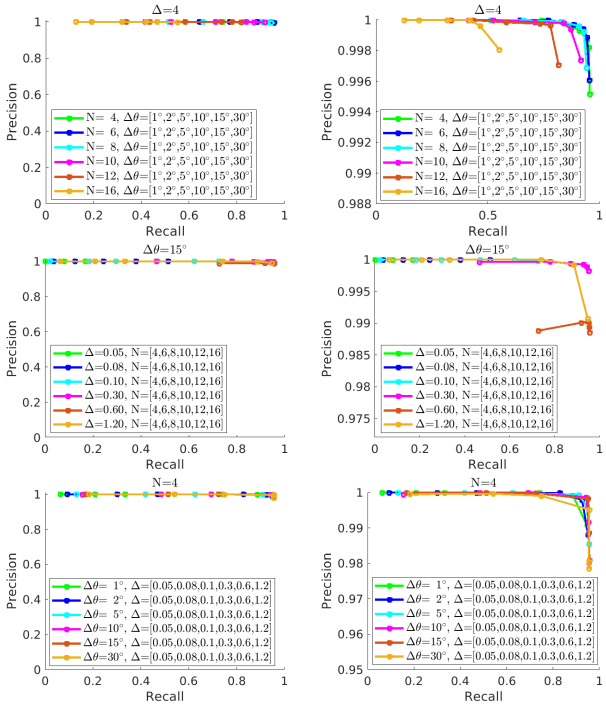

Figure 15. The precision-recall curve for varying values of Δ, $\Delta\theta$, and N (**Left**) and their zoomed-in views (**Right**). These plots suggest an operating point around $\Delta = 0.3$, $\Delta\theta = 15^\circ$, and $N = 4$, which is used in all our experiments, showing the the proposed 3D edge sketch is insensitive to the hyperparameters.

8. Conclusion

This paper advocates for a "reconstruct and then group" approach in contrast to the 3D curve sketch, which advocated for a "group and then reconstruct" approach. The key challenge in this approach is a combinatorial explosion in forming hypotheses between two sets of thousands of edges. The paper successfully addresses this challenge and demonstrates a practical approach to the recovery of the vast majority of curvilinear presentations of a scene. It is expected that a subsequent grouping of the 3D edges would lead to high recall and high precision recovery of 3D curves.

Acknowledgment. The support of NSF award 2312745 is gratefully acknowledged.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016. 7

[2] Dena Bazazian and M Eulàlia Parés. EDC-Net: Edge detection capsule network for 3D point clouds. *Applied Sciences*, 11(4):1833, 2021. 2

[3] Andrea Bignoli, Andrea Romanoni, Matteo Matteucci, and Politecnico di Milano. Multi-view stereo 3D edge reconstruction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 867–875. IEEE, 2018. 2, 7

[4] Mengqi Chen, Laishui Zhou, Honghua Chen, and Jun Wang. Thin-walled aircraft panel edge extraction from 3D measurement surfaces via feature-aware displacement learning. *IEEE Transactions on Instrumentation and Measurement*, 2024. 2

[5] Kseniya Cherenkova, Elona Dupont, Anis Kacem, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Sepicnet: Sharp edges recovery by parametric inference of curves in 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2726–2734, 2023. 2

[6] Ricardo Fabbri and Benjamin Kimia. 3d curve sketch: Flexible curve-based stereo reconstruction and calibration. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1538–1545. IEEE, 2010. 1, 2, 3, 7, 8

[7] Rao Fu, Kai Hormann, and Pierre Alliez. LFS-aware surface reconstruction from unoriented 3D point clouds. *arXiv preprint arXiv:2403.13924*, 2024. 2

[8] Mercedes Garcia-Salguero and Javier Gonzalez-Jimenez. Certifiable solver for real-time n-view triangulation. *IEEE Robotics and Automation Letters*, 8(4):1999–2005, 2023. 6

[9] Yuliang Guo, Naman Kumar, Maruthi Narayanan, and Benjamin Kimia. A multi-stage approach to curve extraction. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 663–678. Springer, 2014. 2

[10] Timo Hackel, Jan D Wegner, and Konrad Schindler. Contour detection in unstructured 3D point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1610–1618, 2016. 2

[11] Markus Herb, Tobias Weiherer, Nassir Navab, and Federico Tombari. Crowd-sourced semantic edge mapping for autonomous vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7047–7053. IEEE, 2019. 2

[12] Chems-Eddine Himeur, Thibault Lejemble, Thomas Pellegrini, Mathias Paulin, Loic Barthe, and Nicolas Mellado. PCEDNet: A lightweight neural network for fast and interactive edge detection in 3D point clouds. *ACM Transactions on Graphics (TOG)*, 41(1):1–21, 2021. 2

[13] Kai-Wen Hsiao, Jia-Bin Huang, and Hung-Kuo Chu. Multi-view wire art. *ACM Trans. Graph.*, 37(6):242, 2018. 2

[14] Shangfeng Huang, Ruisheng Wang, Bo Guo, and Hongxin Yang. PBWR: Parametric building wireframe reconstruction from aerial lidar point clouds. *arXiv preprint arXiv:2311.12062*, 2023. 2

[15] Xianhe Jiao, Chenlei Lv, Ran Yi, Junli Zhao, Zhenkuan Pan, Zhongke Wu, and Yong-Jin Liu. MSL-Net: Sharp feature detection network for 3D point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2

[16] Benjamin B Kimia, Xiaoyan Li, Yuliang Guo, and Amir Tamrakar. Differential geometry in edge detection: accurate estimation of position, orientation and curvature. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1573–1586, 2018. 2, 4, 7

[17] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2, 7, 8

[18] Maxime Lhuillier. Improving thin structures in surface reconstruction from sparse point cloud. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[19] Lei Li, Songyou Peng, Zehao Yu, Shaohui Liu, Rémi Pautrat, Xiaochuan Yin, and Marc Pollefeys. 3D neural edge reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21219–21229, 2024. 2, 6, 7, 8

[20] Shiwei Li, Yao Yao, Tian Fang, and Long Quan. Reconstructing thin structures of manifold surfaces by integrating spatial curves. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2896, 2018. 2

[21] Yangbin Lin, Cheng Wang, Jun Cheng, Bili Chen, Fukai Jia, Zhonggui Chen, and Jonathan Li. Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102:172–183, 2015. 2

[22] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5987–5997, 2021. 1

[23] Lingjie Liu, Nenglun Chen, Duygu Ceylan, Christian Theobalt, Wenping Wang, and Niloy J Mitra. CurveFusion: reconstructing thin structures from RGBD sequences. *arXiv preprint arXiv:2107.05284*, 2021. 2

[24] Yujia Liu, Stefano D'Aronco, Konrad Schindler, and Jan Dirk Wegner. Pc2wf: 3d wireframe reconstruction from raw point clouds. *arXiv preprint arXiv:2103.02766*, 2021. 2

[25] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3D shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pages 67–77. IEEE, 2017. 2

[26] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December*

4, 2016, Revised Selected Papers 1, pages 60–74. Springer, 2017. 2

[27] Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 155–164, 2019. 2

[28] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1923–1932, 2020. 4, 7

[29] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5117–5127, 2021. 4, 7

[30] Anil Usumezbas, Ricardo Fabbri, and Benjamin B Kimia. From multiview image curves to 3D drawings. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 70–87. Springer, 2016. 2, 3

[31] Anil Usumezbas, Ricardo Fabbri, and Benjamin B Kimia. The surfacing of multiview 3d drawings via lofting and occlusion reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2980–2989, 2017. 2, 3

[32] Peng Wang, Lingjie Liu, Nenglun Chen, Hung-Kuo Chu, Christian Theobalt, and Wenping Wang. Vid2Curve: simultaneous camera motion estimation and thin structure reconstruction from an RGB video. *ACM Transactions on Graphics (TOG)*, 39(4):132–1, 2020. 2

[33] Weiran Wang, Huijun Di, and Lingxiao Song. Reconstructing 3D contour models of general scenes from RGB-D sequences. In *International Conference on Multimedia Modeling*, pages 158–170. Springer, 2022. 2

[34] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. *Advances in neural information processing systems*, 33:20167–20178, 2020. 2

[35] Shaobo Xia and Ruisheng Wang. A fast edge extraction method for mobile LiDAR point clouds. *IEEE Geoscience and Remote Sensing Letters*, 14(8):1288–1292, 2017. 2

[36] Yunfan Ye, Renjiao Yi, Zhirui Gao, Chenyang Zhu, Zhiping Cai, and Kai Xu. NEF: Neural edge fields for 3d parametric curve reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8486–8495, 2023. 2, 6, 7, 8

[37] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018. 2

[38] Jie Zhang, Junjie Cao, Xiuping Liu, He Chen, Bo Li, and Ligang Liu. Multi-normal estimation via pair consistency voting. *IEEE transactions on visualization and computer graphics*, 25(4):1693–1706, 2018. 2

[39] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. NerVE: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13601–13610, 2023. 2